

REMARKS

Responsive to the Office Action mailed August 9, 2004, Applicant has studied the Examiner's comments and the cited art. Claims 1-54 are currently pending and remain pending after entry of the amendments. In view of the following remarks, Applicant respectfully submits that the application is in condition for allowance.

Request for Interview Prior to Formal Action on Amendment

Applicant requested an interview prior to submission of the enclosed RCE request and previously faxed a draft amendment to the Examiner for review. The Examiner did not agree to an interview prior to submission of an RCE request, but was agreeable to such an interview after filing of the RCE request. Applicant requests such an interview prior to formal action on this amendment. An "Applicant Initiated Interview Request Form" accompanies this RCE request. Please contact Applicant's undersigned representative to schedule the interview.

Amendments

Applicant has amended the Specification to resolve certain informalities identified by the Office Action. The amendments do not add new matter.

Applicant has amended claims 3, 19, 21, 27, 33, and 42 to resolve issues of antecedent basis identified by the Office Action. The amendments do not narrow the scope of the claims.

Applicant has amended claims 1, 7, 27, 42, and 49 to clarify that the inspector objects are separate from the application objects with which they communicate. The amendment merely makes explicit what was already implicit in the claim and has been previously argued by Applicant as a feature of the claim. Therefore, the amendments do not narrow the scope of the claims. The amendments are supported by at least the following text portions of the Specification:

The document server and inspectors can act as part of the application or as a totally separate application communicating to the runtime applications

through commonly available distributed messaging mechanisms such as CORBA, COM, RPC, and DDE.¹

As can be appreciated the present invention provides an inspector object, which is external to the software application, and is configured to access the objects within the software application in order to obtain and modify attributes thereof.²

Therefore, the amendments do not introduce new matter.

Specification Objections

The Office Action objects to certain informalities in the Specification. Applicant has amended the Specification by replacing the first paragraph of line 7 to resolve the informality identified by the Office Action. For this reason, Applicant respectfully requests withdrawal of the objection.

Claim Rejections Under 35 U.S.C. § 112, 2nd paragraph

Claims 3, 19–21, 25–41, and 42–48 are rejected under 35 U.S.C § 112, 2nd paragraph as indefinite, because of issues of antecedent basis.

Applicant has amended claims 3, 19, 21, 27, 33, and 42 to resolve the issues of antecedent basis identified by the Office Action. For these reasons, Applicant respectfully requests withdrawal of the rejections.

Claims 20, 25–26, 28–41, and 43–48 are rejected for dependency upon rejected base claims 3, 27, and 42, which are now allowable. Claims 20, 25–26, 28–41, and 43–48 are therefore also allowable. For at least this reason, Applicant respectfully requests withdrawal of the rejections.

¹ Applicant does not yet have access to the USPTO's version of the originally filed patent application. This text corresponds to page 4, lines 1-3 of Applicant's electronic text version which may not exactly match the page and line numbers of the USPTO's version of the originally filed patent application.

² Corresponds to page 8, lines 1-3 of Applicant's electronic text version of the patent application.

Claim Rejections Under 35 U.S.C. § 102

Claims 1–54 are rejected under 35 U.S.C. § 102(e) as being anticipated by Faustini, U.S. Patent No. 5,842,020. Applicants respectfully traverse the rejections.

1. Independent claims 1 and 7

With respect to independent claims 1 and 7, the Office Action generally mischaracterizes Faustini and fails to appreciate the difference between the predefined editor built into each edit-capable component of Faustini, which can only edit the object into which the editor is built, and the inspector of Applicant’s claimed subject matter, which is not built-in and integral to the edited objects in the execution environment, but is separate from and capable of communicating with other objects in the execution environment. Faustini recites a built-in editor capability in edit-capable objects that is based on a component-specific editor method. The component-specific editor method is inherited from a template used to define the object, which can be customized to provide features specific to the object. This built-in editor method of an object is provided “as an integral part of the class template from which it is instantiated.”³

For example, Faustini provides “source code enabling the container editor for modifying the properties of the container.”⁴ This editor applies only to objects of the VJContainer class.⁵ An object that was not of the VJContainer class would require a different editor predefined for the appropriate class. For example, an object of class VJNode (which is a class extended by the class VJContainer) would not have available the containerEditor method, but would need to use a different editor method that would be predefined for the VJNode class. VJVScrollbar objects have a vscrollbarEditor method specific to that kind of object⁶ and VJLabel objects have a labelEditor method specific to VJLabel objects, where each editor method is separately predefined by the template for the object class. A VJLabel object cannot use the vscrollbarEditor

³ Col. 153, line 58–Col. 154, line 2.

⁴ Col. 33, lines 1–2; Col. 35 *et seq.*

⁵ Cols. 11–32.

⁶ Cols. 147–150.

method to edit a label, nor can a VJVScrollbar object use the labelEditor method to edit the scrollbar. Thus, Faustini recites editors as methods of the object class, which are built-in and integral to the objects. In addition, because each object requires an editor method specific for the object class, if no editor method is predefined for that object class, objects of that class are not edit-capable.

In contrast, Applicant's claimed subject matter recites an inspector that is "separate from the application object" and that is capable of "communicating information pertaining to the attributes of the application object while the application object is deployed in the execution environment." This inspector is not a method of the application object as in Faustini, but is a separate entity of the object-oriented software application system. As recited by Applicant, an inspector may be capable of communicating information pertaining to the attributes of different types of objects, without the need for a predefined editor specific to that object class. Thus, the inspector is separate from the application object with which it communicates, unlike the built-in editor of the objects of Faustini. For these reasons, Applicant respectfully requests withdrawal of the rejections.

In addition, contrary to the assertion of the Office Action, Faustini fails to recite a "document server" as in Applicant's claimed subject matter. The passage of Faustini recited by the Office Action refers to the built-in editor of the object itself, and thus cannot recite a document server as in Applicant's claimed subject matter. Furthermore, the cited passage merely states that the built-in editor method is automatically invoked upon instantiation of the object that includes that built-in editor method. Even if the edit window 1104 of Faustini that may be created by the editor method is a user interface to the editor method, which Applicant does not admit, the Office Action fails to assert that Faustini recites a document server for maintaining an inventory of objects deployed in the execution environment, or that this document server provides a user interface to the inspector of Applicant's claimed subject matter.

Nothing in Faustini recites or even suggests that the editor method built into Faustini's objects acts as a document server, or that the editor method maintains an inventory of objects deployed in the execution environment. The Office Action cannot simply ignore these elements of Applicant's claimed subject matter. "Anticipation requires the presence in a single prior art

reference disclosure of each and every element of the claimed invention, arranged as in the claim.”⁷ Because the Office Action fails to assert the presence in Faustini of all of the elements of Applicant’s claimed subject matter, Faustini cannot anticipate the claims. For these additional reasons, Applicant respectfully requests withdrawal of the rejections.

2. Independent Claim 27

With respect to independent claim 27, the Office Action does not make specific assertions regarding Applicant’s claimed subject matter other than to cite “column 10, lines 13-51 and column 154, lines 5-8” and “column 154, lines 8-35.”⁸ As Applicant has shown above, the built-in editor methods recited by Faustini in these passages do not recite independent inspector documents that are separate from the application objects of Applicant’s claimed subject matter. For these reasons, Applicant respectfully requests withdrawal of the rejections.

In addition, Faustini fails to recite, either in the cited passages or elsewhere, “a document server process configured for accessing a registry of executable objects launched into the execution environment.” Faustini recites a VJ Tool applet that is initialized in a Java Run-Time Environment 306. Faustini fails to recite the Java Run-Time Environment 306 as providing a “registry of executable objects launched into the execution environment.” Nor does the Java Run-Time Environment 306 act as a document server. Faustini also fails to recite that the VJ Tool applet performs these functions.

Furthermore, the cited passage of Faustini fails to recite “a plurality of inspector documents each comprising an archived inspector configured to alter an attribute of an executable object.” The built-in editor method of the objects of Faustini are not “inspector documents.” Nor are the editor methods of Faustini separately archived, because they are integral with the objects with which the editor methods are associated.

Additionally, the editor methods of Faustini are not deployed by a “user-operable means” of the document server process into the execution environment “based on a registry and a corresponding archived inspector.” Instead, Faustini’s editor methods are invoked automatically

⁷ *Lindemann Maschinenfabrik v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1458, 221 U.S.P.Q. 481, 485 (Fed. Cir. 1984).

⁸ Paper 20040723, p. 15.

upon instantiation, i.e., without user intervention or even the ability to prevent such invocation.⁹ Therefore, Faustini fails to recite either the use of a registry, a user-operable means for deploying, or a corresponding archived inspector. For these additional reasons, Applicant respectfully requests withdrawal of the rejections.

3. Independent Claims 42 and 49

As with claim 7, the Office Action's rejection of claim 42 is simply a list of citations to Faustini, with no explanatory text. The rejection of claim 49 asserts that claim 49 is "another version[]" of claim 42, and that all of the arguments against claim 49 can be found in the discussion of claim 42. The passages of Faustini cited by the Office Action, contrary to the Office Action's assertion, do not recite Applicant's claimed subject matter.

As shown above, Faustini fails to recite an inspector separate from the object being modified by the inspector. Applicant's arguments to this effect above are incorporated herein by reference.

In addition, Faustini fails to recite a "selection means for selecting an inspector configured for modification of a corresponding application object." The editor window and editor method predefined in the class templates for the objects of Faustini are not selected by any selection means, but are automatically invoked upon instantiation of the object, as Applicant has shown above. The editor capability is "added ... to each class template for which an editing capability is desired in component objects instantiated therefrom."¹⁰ Faustini does not recite selecting between multiple editors; rather, the single predefined editor for the object class is always invoked upon instantiation of the object.¹¹ For these reasons, Applicant respectfully requests withdrawal of the rejections.

⁹ Col. 154, lines 5–8.

¹⁰ Col. 154, lines 14–16.

¹¹ Col. 154, lines 5–8.

4. Dependent Claims

Claims 2–6, 8–26, 28–41, 43–48, and 50–53 depend from allowable independent claims 1, 7, 27, 42, and 49, and are therefore also allowable. For at least this reason, Applicant respectfully requests withdrawal of the rejections.

In addition, with respect to claim 2, even if the editor methods of Faustini correspond to the inspector of Applicant's claimed subject matter, which Applicant denies, Faustini recites that the editor methods are invoked by the object upon instantiation, not by actuation of another entity in the system. In contrast, Applicant's claimed subject matter state that "the document server is configured to actuate the inspector." As shown above, Faustini recites no such document server. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 5, Faustini fails to recite a document server that "retrieve[s] and instantiate[s] a corresponding inspector for a deployed object desired to be modified." As shown above, the editor method of an object is integral with the object and is not separately instantiated in the execution environment of Faustini. Rather, the editor method is automatically invoked upon instantiation of the object itself. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 12, Faustini fails to recite an editor method that can be "configured to modify [a] visual attribute to be conditionally dependent on a characteristic of the execution environment." The passage cited by the Office Action is directed to using the VJ Tool application to add text to scrollbars by adding label text fields to the scrollbar, there the label text is chosen by the user. Faustini does not, either in the cited passage or elsewhere, recite conditional attribute values, much less using the editor to set an attribute of the object to a value that is conditionally dependent on a characteristic of the execution environment. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claims 14 and 17, the Office Action mischaracterizes the cited passage of Faustini as disclosing "wherein said inspector is provided in the form of an archive file accessible to the document server." As shown above, Faustini clearly recites the editor as integral to the component with which it is associated. The cited passage does not recite an editor

method as being provided “in the form of an archive file,” but merely recites the applet or application “can be obtained” by a web browser. Because Applicant’s inspector is separate from the application object with which it communicates, the inspector object can be separately archived and de-archived as desired. In contrast, Faustini’s editor method inherently is loaded with the object itself. For these additional reasons, Applicant respectfully requests withdrawal of the rejections.

With respect to claims 15, 18, 41, and 54, the Office Action mischaracterizes the cited passage of Faustini as disclosing that the archive file is an XML file. Faustini nowhere mentions the existence of XML files, much less recites an archive file as being in that format. For these additional reasons, Applicant respectfully requests withdrawal of the rejections.

With respect to claim 19, the Office Action mischaracterizes Faustini by asserting that the editor method that has been customized for the specific object with which it is associated is selected on the basis of “object attribute information contained within said inventory.” First, Faustini fails to recite an object inventory that contains object attributes. Second, even if Faustini recites such an inventory, which Applicant denies, Faustini fails to recite selecting an editor method based on object attribute information contained within the inventory. Instead, as shown above, the editor methods of Faustini are not selected at runtime, but are predetermined and built into the object, and automatically invoked when the object is instantiated. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claims 21 and 22, Faustini fails to recite “searching and identifying each object deployed in said execution environment having a common attribute selected for modification, to allow selective modification of each such object”¹² or “searching a library of archived objects, and for identifying archived objects having a common attribute selected for modification, to allow selective modification of each object.”¹³ The passages of Faustini cited without explanation or comment by the Office Action do not recite such functionality. Faustini does not recite the VJ Tool application generally described in Col. 10 of Faustini as having the ability to search for or otherwise identify every object that has a selected common attribute,

¹² Claim 21.

¹³ Claim 22.

whether the object is deployed in the VJ Tool execution environment or is saved in a library. Similarly, the editor methods and windows built into the components recited in Col. 154 do not provide this functionality. The only selection function recited by Faustini for selecting multiple objects is a “select all” function, which clearly does not select only objects having a common attribute.

Instead, as shown above, Faustini’s objects each contain their own editor methods, where “the editor is customized for the specific component with which it is associated.”¹⁴ The editor “appears when a component is instantiated,” if an editor has been predetermined and predefined for that class of object.¹⁵

Furthermore, because the editor opens automatically on instantiation of the object, Faustini does not recite selectively modifying objects with common attributes, but requires the user to individually edit each such object. For these additional reasons, Applicant respectfully requests withdrawal of the rejections.

With respect to claim 23, Faustini fails to recite, either in the passage cited by the Office Action or elsewhere, that any version tracking or history of modifications is provided by VJ Tool. The mere recitation by Faustini that VJ Tool is a development tool does not inherently imply such functionality. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 24, as shown above, the editor for a given object of Faustini is built-in and integral to the component with which it is associated. Faustini nowhere recites any register for “selecting the inspector corresponding to the runtime object.” No such register would be required, because the editor is not selected at run time, but is built-in and predetermined for each object. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 38, Faustini fails to recite “select[ing] among alternative archived inspectors according to an operating system of the execution environment.” First, as shown above, Faustini recites predetermined and built-in editors for edit capable objects. Faustini

¹⁴ Col. 154, lines 39-41.

¹⁵ Col. 154, line 41; Col. 153, line 58-Col. 154, line 11.

nowhere suggests selecting an alternative editor, but only predefines a single editor per object. Second, the passages cited by the Office Action without further explanation or comment do not refer to alternative editors, but merely recite the functionality of Faustini as involving a Java Run-Time Environment. Faustini's use of Java suggests an application that can be run without change upon "any machine with a bytecode interpreter,"¹⁶ which is contrary to the concept of varying and selecting the inspectors "according to an operating system of the execution environment." For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 39, Faustini does not recite a generic inspector as in Applicant's claimed subject matter. To the contrary, Faustini recites that an editor must be defined for each edit capable object.¹⁷ "[T]he editor is customized for the specific component with which it is associated."¹⁸ Such a customized editor would not be understood as generic by one of ordinary skill in the art. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

With respect to claim 44, Faustini fails to recite two execution environments that are different operating systems. Instead, contrary to the Office Action's assertion, Faustini recites a system limited to a Java Run-Time Environment, which is a runtime environment that attempts to present an operating system independent execution environment for Java programs. For these additional reasons, Applicant respectfully requests withdrawal of the rejection.

Conclusion

Applicant respectfully submits that all issues and rejections have been adequately addressed, that all claims as amended are allowable, and that the case should be advanced to issuance.

If the Examiner has any questions or wishes to discuss the claims, Applicant encourages the Examiner to call the undersigned at the telephone number indicated below.

¹⁶ Col. 9, lines 27-28.

¹⁷ Col. 153, line 58-Col. 154, line 11.

¹⁸ Col. 154, lines 39-41.

Respectfully submitted,

ROBERT L. SCHMITTER

February 9, 2005
(Date)

By:

Clark Jablon

CLARK A. JABLON

Registration No. 35,039

AKIN GUMP STRAUSS HAUER & FELD LLP

One Commerce Square

2005 Market Street - Suite 2200

Philadelphia, PA 19103

Direct Dial: (215) 965-1293

Facsimile: (215) 965-1210

Customer No. 000570

Enclosure – RCE Request